

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ.....	2
ФОРМУЛИРОВКА ЗАДАНИЙ.....	3
ВЫПОЛНЕНИЕ ЗАДАНИЙ.....	6
Часть 1.....	6
Часть 2.....	11
Часть 3.....	17
КОНТРОЛЬНЫЕ ВОПРОСЫ.....	27
ВЫВОД.....	29

ФОРМУЛИРОВКА ЗАДАНИЙ

Цель работы: получить навыки по работе с командной строкой и git'ом.

Часть 1. Основные команды Git

1. Установите и настройте клиент git на своей рабочей станции.
2. Создайте локальный репозиторий и добавьте в него несколько файлов.
3. Внесите изменения в один из файлов.
4. Проиндексируйте изменения и проверьте состояние.
5. Сделайте коммит того, что было проиндексировано в репозиторий.
Добавьте к коммиту комментарий.

6. Измените еще один файл. Добавьте это изменение в индекс git. Измените файл еще раз. Проверьте состояние и произведите коммит проиндексированного изменения.

Теперь добавьте второе изменение в индекс, а затем проверьте состояние с помощью команды `git status`. Сделайте коммит второго изменения.

7. Просмотрите историю коммитов с помощью команды `git log`. Ознакомьтесь с параметрами команды и используйте некоторые из них для различного формата отображения истории коммитов.

8. Верните рабочий каталог к одному из предыдущих состояний.
9. Изучите, как создавать теги для коммитов для использования в будущем.
10. Отмените некоторые изменения в рабочем каталоге (до и после индексирования).

11. Отмените один из коммитов в локальном репозитории.

Часть 2. Системы управления репозиториями

1. Создайте аккаунт на GitHub (у кого нет),

2. Создайте репозиторий на GitHub и на локальной машине, согласно выбранной теме проекта,
3. Создайте несколько файлов на локальной машине при помощи консоли,
4. Создайте SSH-ключ для авторизации,
5. Свяжите репозиторий локальной машины с репозиторием на GitHub при помощи консоли,
6. Создайте новую ветку в репозитории с помощью команды, произведите в ней какие-нибудь изменения, а после слейте с веткой master,
7. Выполните цепочку действий в репозитории, согласно вариантам.

№ Варианта	Задание
2	1) Клонировать непустой удаленный репозиторий на локальную машину 2) Создайте новую ветку и выведите список всех веток 3) Произведите коммит в ветке master 4) Произведите 3 коммита в новой ветке в разные файлы 5) Выгрузите изменения в удаленный репозиторий 6) Откатите ветку обратно на 2 коммита (в том числе в удаленном репозитории) 7) Выведите в консоли различия между веткой master и новой веткой 8) Перебазировать новую ветку на master

Часть 3. Работа с ветвлением и оформление кода

1. Сделайте форк репозитория в соответствии с вашим вариантом
2. Склонируйте его на локальную машину
3. Создайте две ветки branch1 и branch2 от последнего коммита в master'e
4. Проведите по 3 коммита в каждую из веток, которые меняют один и тот же кусочек файла

5. Выполните слияние ветки branch1 в ветку branch2, разрешив конфликты при этом
6. Выгрузите все изменения во всех ветках в удаленный репозиторий
7. Проведите еще 3 коммита в ветку branch1
8. Склонируйте репозиторий еще раз в другую директорию
9. В новом клоне репозитории сделайте 3 коммита в ветку branch1
10. Выгрузите все изменения из нового репозитория в удаленный репозиторий
11. Вернитесь в старый клон с репозиторием, выгрузите изменения с опцией --force
12. Получите все изменения в новом репозитории

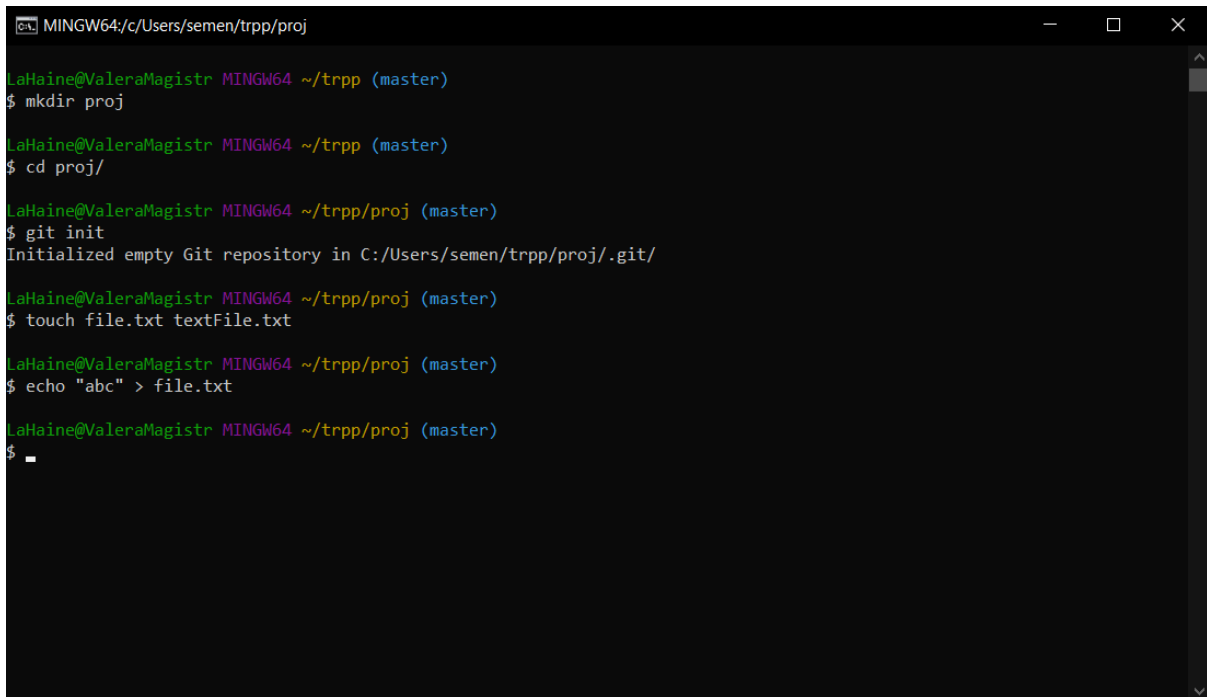
№ Варианта	Задание
2	https://github.com/google/model_search

ВЫПОЛНЕНИЕ ЗАДАНИЙ

Часть 1

Клиент git был установлен и настроен.

Создание локального репозитория с несколькими файлами, и внесение изменений в один из них.



```
MINGW64:/c/Users/semen/trpp/proj
LaHaine@ValeraMagistr MINGW64 ~/trpp (master)
$ mkdir proj
LaHaine@ValeraMagistr MINGW64 ~/trpp (master)
$ cd proj/
LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git init
Initialized empty Git repository in C:/Users/semen/trpp/proj/.git/
LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ touch file.txt textFile.txt
LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ echo "abc" > file.txt
LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$
```

Рисунок 1 – Выполнение заданий 1-3 первой части

Индексирование изменений и проверка состояния, коммит проиндексированных изменений и добавление комментария.

```
MINGW64/c/Users/semn/trpp/proj
LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ echo "abc" > file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git add file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git commit -m "Commit"
[master (root-commit) f4366de] Commit
 1 file changed, 1 insertion(+)
 create mode 100644 file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$
```

Рисунок 2 – Выполнение заданий 3-5 первой части

Изменение файла, индексирование, изменение файла повторно с проверкой состояния. Создание коммита с изменением. Индексирование второго изменения с проверкой состояния и последующим коммитом.

```
MINGW64/c/Users/semn/trpp/proj
LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ echo "abcd" > textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git add textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ echo "123" > textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git status
On branch master

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   textFile.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git commit -m "new commit"
[master e291e3e] new commit
 1 file changed, 1 insertion(+)
 create mode 100644 textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$
```

Рисунок 3 – Выполнение задания 6 первой части

```
MINGW64/c/Users/semn/trpp/proj
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git commit -m "new commit"
[master e291e3e] new commit
1 file changed, 1 insertion(+)
create mode 100644 textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git add textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git commit -m "new commit"
[master fc776d8] new commit
1 file changed, 1 insertion(+), 1 deletion(-)

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$
```

Рисунок 4 – Выполнение задания 6 первой части

Просмотр истории коммитов, использование различных форм для отображения истории коммитов.

```
MINGW64/c/Users/semn/trpp/proj
1 file changed, 1 insertion(+), 1 deletion(-)

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git log
commit fc776d85ea37de90f72eb90577a1ead6683d4196 (HEAD -> master)
Author: sgolev <dokoeboi@gmail.com>
Date: Thu Feb 27 01:14:05 2025 +0300

    new commit

commit e291e3e797bed02c23bfac95995d98a1d6acdc8b
Author: sgolev <dokoeboi@gmail.com>
Date: Thu Feb 27 01:12:16 2025 +0300

    new commit

commit f4366deba0c72836402b3c02ac2846b37deb4359
Author: sgolev <dokoeboi@gmail.com>
Date: Thu Feb 27 01:07:16 2025 +0300

    Commit

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git log --oneline
fc776d8 (HEAD -> master) new commit
e291e3e new commit
f4366de Commit

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$
```

Рисунок 5 – Выполнение задания 7 первой части

```
MINGW64/c/Users/semen/trpp/proj
Commit

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git log --oneline
fc776d8 (HEAD -> master) new commit
e291e3e new commit
f4366de Commit

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git log --graph
* commit fc776d85ea37de90f72eb90577a1ead6683d4196 (HEAD -> master)
  Author: sgolev <dokoeboi@gmail.com>
  Date: Thu Feb 27 01:14:05 2025 +0300

    new commit

* commit e291e3e797bed02c23bfac95995d98a1d6acdc8b
  Author: sgolev <dokoeboi@gmail.com>
  Date: Thu Feb 27 01:12:16 2025 +0300

    new commit

* commit f4366deba0c72836402b3c02ac2846b37deb4359
  Author: sgolev <dokoeboi@gmail.com>
  Date: Thu Feb 27 01:07:16 2025 +0300

    Commit

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$
```

Рисунок 6 – Выполнение задания 7 первой части

Возвращение рабочего каталога к состоянию на момент второго коммита (хэш : e291e3e).

```
MINGW64/c/Users/semen/trpp/proj

* commit f4366deba0c72836402b3c02ac2846b37deb4359
  Author: sgolev <dokoeboi@gmail.com>
  Date: Thu Feb 27 01:07:16 2025 +0300

    Commit

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git checkout e291e3e
Note: switching to 'e291e3e'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

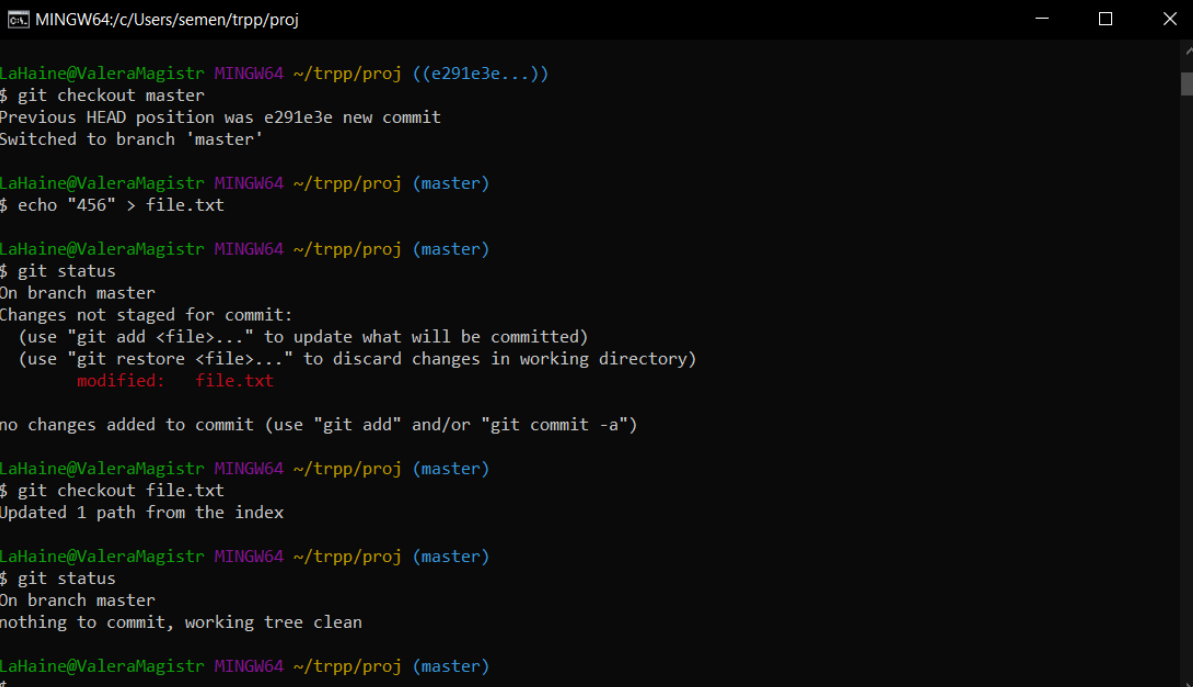
HEAD is now at e291e3e new commit

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj ((e291e3e...))
$
```

Рисунок 7 – Выполнение задания 8 первой части

Изучено создание тегов.

Внесение изменений в файл, отменим это изменение, далее произведены те же действия, но с индексацией изменений.



```
MINGW64/c/Users/semen/trpp/proj
LaHaine@ValeraMagistr MINGW64 ~/trpp/proj ((e291e3e...))
$ git checkout master
Previous HEAD position was e291e3e new commit
Switched to branch 'master'

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ echo "456" > file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file.txt

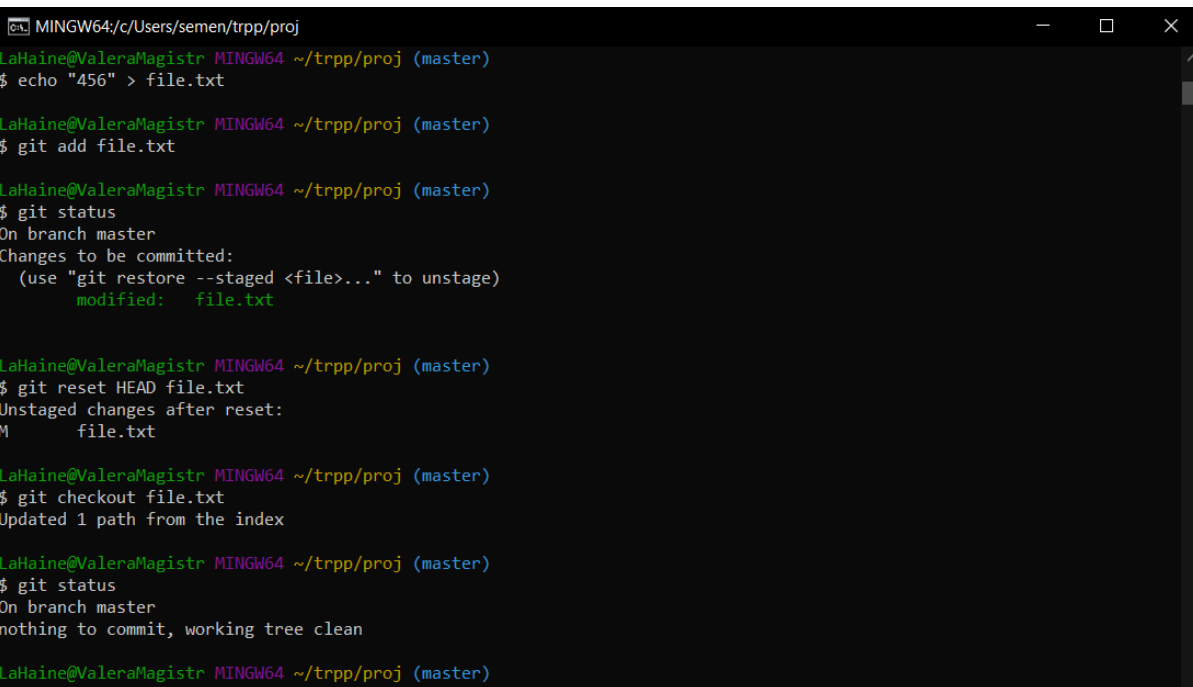
no changes added to commit (use "git add" and/or "git commit -a")

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git checkout file.txt
Updated 1 path from the index

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git status
On branch master
nothing to commit, working tree clean

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$
```

Рисунок 8 – Выполнение задания 10 первой части



```
MINGW64/c/Users/semen/trpp/proj
LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ echo "456" > file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git add file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git reset HEAD file.txt
Unstaged changes after reset:
M       file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git checkout file.txt
Updated 1 path from the index

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git status
On branch master
nothing to commit, working tree clean

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$
```

Рисунок 9 – Выполнение задания 10 первой части

Внесение изменений в файл, индексирование и создания коммита с изменениями. Далее отмена этого коммита.

```
MINGW64/c/Users/semen/trpp/proj
On branch master
nothing to commit, working tree clean

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ echo "789" > textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git add .

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git commit -m "wrong commit"
[master 5d67baf] wrong commit
1 file changed, 1 insertion(+), 1 deletion(-)

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git revert HEAD --no-edit
[master f1d7348] Revert "wrong commit"
Date: Thu Feb 27 01:39:09 2025 +0300
1 file changed, 1 insertion(+), 1 deletion(-)

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$ git log --oneline
f1d7348 (HEAD -> master) Revert "wrong commit"
5d67baf wrong commit
fc776d8 new commit
e291e3e new commit
f4366de Commit

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj (master)
$
```

Рисунок 10 – Выполнение задания 11 первой части

Часть 2

Аккаунт на GitHub был создан, также создан локальный репозиторий с парой файлов.

```
MINGW64/c/Users/semen/trpp/proj2

LaHaine@ValeraMagistr MINGW64 ~/trpp (master)
$ mkdir proj2

LaHaine@ValeraMagistr MINGW64 ~/trpp (master)
$ cd proj2

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj2 (master)
$ touch file.txt textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj2 (master)
$ git init
Initialized empty Git repository in C:/Users/semen/trpp/proj2/.git/

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj2 (master)
$
```

Рисунок 11 – Выполнение заданий 1-3 второй части

Создан репозиторий на GitHub.

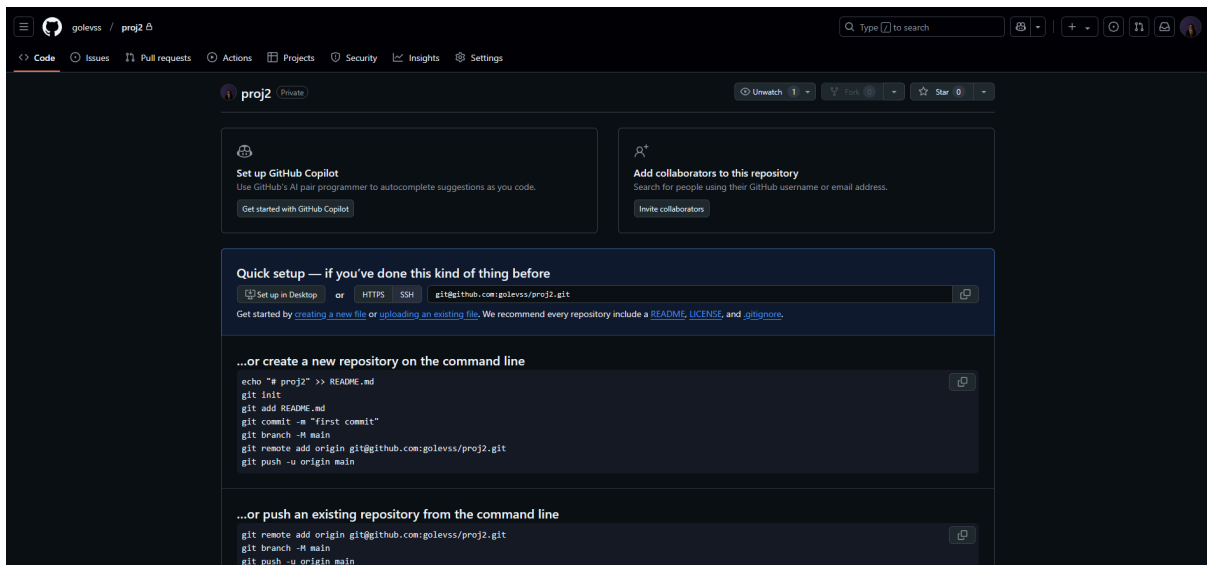


Рисунок 12 – Выполнение задания 2 второй части

Связывание локального репозитория, и репозитория созданного на GitHub.

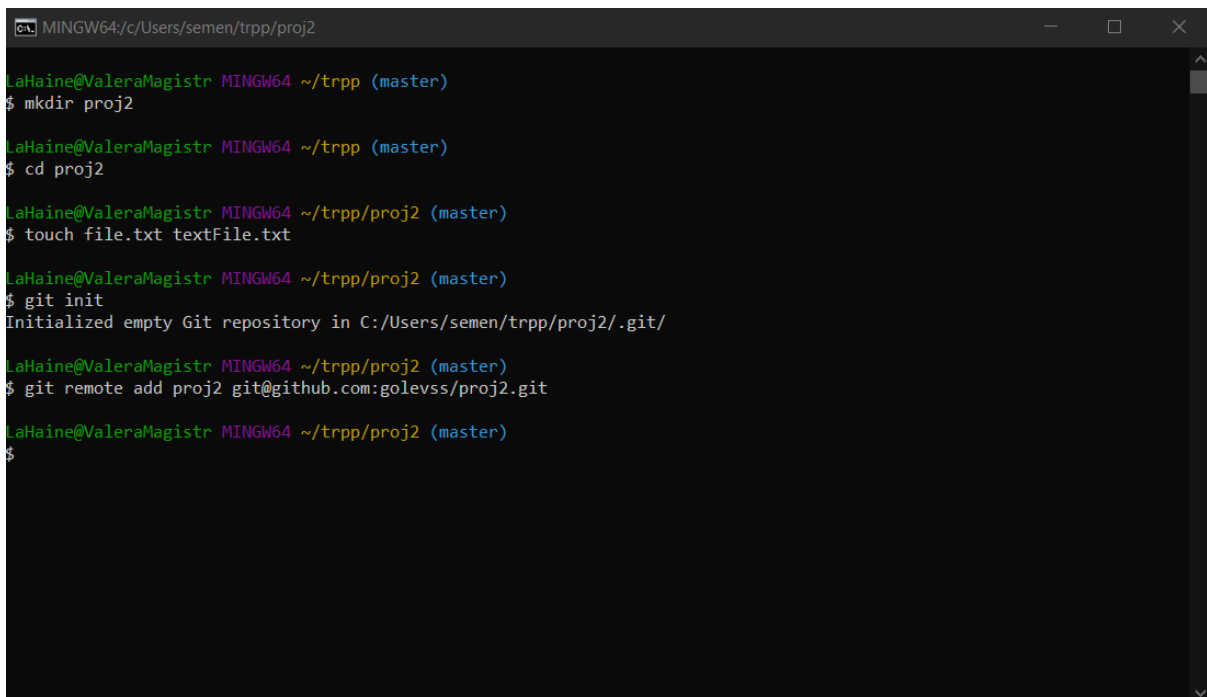


Рисунок 13 – Выполнение заданий 4-5 второй части

Создание новой ветки в репозитории, произведение изменений и сливание данной ветки с веткой master.

```
MINGW64/c/Users/semen/trpp/proj2
$ git remote add proj2 git@github.com:golevss/proj2.git

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj2 (master)
$ git checkout -b dev_branch
Switched to a new branch 'dev_branch'

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj2 (dev_branch)
$ echo "159" > file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj2 (dev_branch)
$ git add file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj2 (dev_branch)
$ git commit -m "commit on new branch"
[dev_branch (root-commit) 9d8c39d] commit on new branch
1 file changed, 1 insertion(+)
create mode 100644 file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj2 (dev_branch)
$ git checkout -b master
Switched to a new branch 'master'

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj2 (master)
$ git merge dev_branch
Already up to date.

LaHaine@ValeraMagistr MINGW64 ~/trpp/proj2 (master)
$
```

Рисунок 14 – Выполнение задания 6 второй части

Клонирование непустого локального репозитория.

```
MINGW64/c/Users/semen/trpp/MIREATOM

LaHaine@ValeraMagistr MINGW64 ~/trpp (new_branch)
$ ls
proj/  proj2/

LaHaine@ValeraMagistr MINGW64 ~/trpp (new_branch)
$ git clone https://github.com:golevss/MIREATOM.git
Cloning into 'MIREATOM'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (11/11), done.
Receiving objects: 100% (13/13), 9.80 KiB | 3.27 MiB/s, done.
Resolving deltas: 100% (2/2), done.
remote: Total 13 (delta 2), reused 12 (delta 1), pack-reused 0 (from 0)

LaHaine@ValeraMagistr MINGW64 ~/trpp (new_branch)
$ cd MIREATOM/

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (master)
$
```

Рисунок 15 – Выполнение пункта 1 из заданий варианта второй части

Создание новой ветки, коммит в ветке master.

```
MINGW64/c/Users/semen/trpp/MIREATOM
LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (master)
$ git checkout -b "new_branch"
Switched to a new branch 'new_branch'

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git branch
  master
* new_branch

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (master)
$ touch file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (master)
$ git add .

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (master)
$ git commit "master commit"
error: pathspec 'master commit' did not match any file(s) known to git

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (master)
$ git commit -m "master commit"
[master 99de857] master commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file.txt
```

Рисунок 16 – Выполнение пункта 2 из заданий варианта второй части
Произведение трёх коммитов в новой ветке в разные файлы.

```
MINGW64/c/Users/semen/trpp/MIREATOM
$ git branch
  master
* new_branch

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ touch textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git add textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git commit -m "New textfile"
[new_branch 8769129] New textfile
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ touch htmlFile.html

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git add htmlFile.html

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git commit -m "New html file"
[new_branch 0200b83] New html file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 htmlFile.html

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$
```

Рисунок 17 – Выполнение пункта 3 из заданий варианта второй части

```
MINGW64/c/Users/semn/trpp/MIREATOM
[new_branch 8769129] New textfile
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ touch htmlFile.html

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git add htmlFile.html

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git commit -m "New html file"
[new_branch 0200b83] New html file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 htmlFile.html

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ touch mdFile.md

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git add mdFile.md

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git commit -m "New MarkDown file"
[new_branch a95794d] New MarkDown file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mdFile.md

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$
```

Рисунок 18 – Выполнение пункта 3 из заданий варианта второй части
Выгрузка изменений в удаленный репозиторий.

```
MINGW64/c/Users/semn/trpp/MIREATOM
[new_branch 0200b83] New html file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 htmlFile.html

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ touch mdFile.md

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git add mdFile.md

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git commit -m "New MarkDown file"
[new_branch a95794d] New MarkDown file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mdFile.md

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git push origin new_branch
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 977 bytes | 977.00 KiB/s, done.
Total 9 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/golevss/MIREATOM.git
  8a6e8d1..a95794d  new_branch -> new_branch

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$
```

Рисунок 18 – Выполнение пункта 4 из заданий варианта второй части
Возвращение ветки на 2 коммита обратно, в том числе в удалённом
репозитории.

```
MINGW64/c/Users/semn/trpp/MIREATOM
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 977 bytes | 977.00 KiB/s, done.
Total 9 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/golevss/MIREATOM.git
  8a6e8d1..a95794d new_branch -> new_branch

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git revert HEAD~2
[new_branch 2b282c3] Revert "New textfile"
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 textFile.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git push origin new_branch
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 268 bytes | 268.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/golevss/MIREATOM.git
  a95794d..2b282c3 new_branch -> new_branch

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$
```

Рисунок 19 – Выполнение пункта 5 из заданий варианта второй части
Вывод различий между двумя ветками.

```
MINGW64/c/Users/semn/trpp/MIREATOM
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 268 bytes | 268.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/golevss/MIREATOM.git
  a95794d..2b282c3 new_branch -> new_branch

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git diff master new_branch
diff --git a/file.txt b/file1.txt
similarity index 100%
rename from file.txt
rename to file1.txt
diff --git a/file2.txt b/file2.txt
new file mode 100644
index 0000000..e69de29
diff --git a/file3.txt b/file3.txt
new file mode 100644
index 0000000..e69de29
diff --git a/htmlFile.html b/htmlFile.html
new file mode 100644
index 0000000..e69de29
diff --git a/mdFile.md b/mdFile.md
new file mode 100644
index 0000000..e69de29

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$
```

Рисунок 20 – Выполнение пункта 6 из заданий варианта второй части
Перебазирование новой ветки на ветку master.

```
MINGW64/c/Users/semn/trpp/MIREATOM
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/golevss/MIREATOM.git
 a95794d..2b282c3 new_branch -> new_branch

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git diff master new_branch
diff --git a/file.txt b/file1.txt
similarity index 100%
rename from file.txt
rename to file1.txt
diff --git a/file2.txt b/file2.txt
new file mode 100644
index 0000000..e69de29
diff --git a/file3.txt b/file3.txt
new file mode 100644
index 0000000..e69de29
diff --git a/htmlFile.html b/htmlFile.html
new file mode 100644
index 0000000..e69de29
diff --git a/mdFile.md b/mdFile.md
new file mode 100644
index 0000000..e69de29

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$ git rebase master
Successfully rebased and updated refs/heads/new_branch.

LaHaine@ValeraMagistr MINGW64 ~/trpp/MIREATOM (new_branch)
$
```

Рисунок 21 – Выполнение пунктов 7-8 из заданий варианта второй части

Часть 3

Сделан форк репозитория указанного в задании варианта.

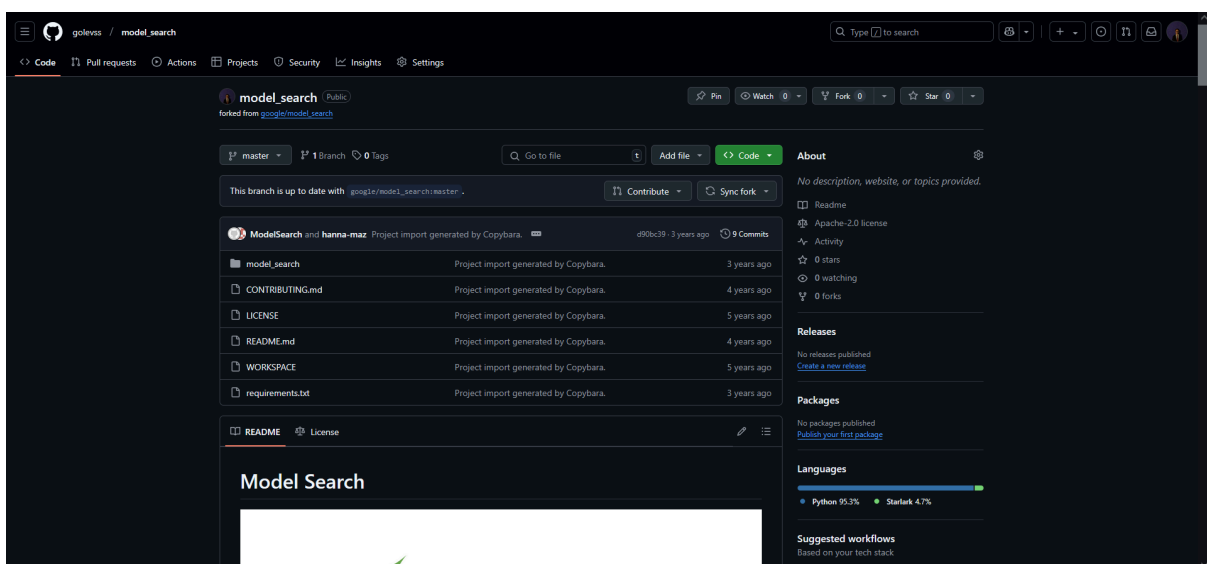


Рисунок 22 – Выполнение задания 1 третьей части

Репозиторий клонирован на локальную машину.


```
MINGW64/c/Users/semen/trpp/model_search

LaHaine@ValeraMagistr MINGW64 ~/trpp (new_branch)
$ git clone https://github.com/golevss/model_search.git
Cloning into 'model_search'...
remote: Enumerating objects: 274, done.
remote: Counting objects: 100% (76/76), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 274 (delta 49), reused 40 (delta 40), pack-reused 198 (from 1)
Receiving objects: 100% (274/274), 291.58 KiB | 1.22 MiB/s, done.
Resolving deltas: 100% (128/128), done.

LaHaine@ValeraMagistr MINGW64 ~/trpp (new_branch)
$ cd model_search/

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (master)
$
```

Рисунок 23 – Выполнение задания 2 третьей части

Создадим две новые ветки и проведём в каждой по три коммита.

```
MINGW64/c/Users/semen/trpp/model_search

LaHaine@ValeraMagistr MINGW64 ~/trpp (new_branch)
$ git clone https://github.com/golevss/model_search.git
Cloning into 'model_search'...
remote: Enumerating objects: 274, done.
remote: Counting objects: 100% (76/76), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 274 (delta 49), reused 40 (delta 40), pack-reused 198 (from 1)
Receiving objects: 100% (274/274), 291.58 KiB | 1.22 MiB/s, done.
Resolving deltas: 100% (128/128), done.

LaHaine@ValeraMagistr MINGW64 ~/trpp (new_branch)
$ cd model_search/

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (master)
$ git checkout -b "branch2"
Switched to a new branch 'branch2'

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git checkout -b "branch1"
Switched to a new branch 'branch1'

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git branch
* branch1
  branch2
  master

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$
```

Рисунок 24 – Выполнение задания 3 третьей части

```
MINGW64/c/Users/semen/trpp/model_search
LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git branch
* branch1
  branch2
  master

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ touch file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ echo "123" > file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git add file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git commit -m "Commit on branch1"
[branch1 8b6f770] Commit on branch1
1 file changed, 1 insertion(+)
create mode 100644 file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git checkout branch2
Switched to branch 'branch2'

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ echo "466" > file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git add .
```

Рисунок 25 – Выполнение задания 4 третьей части

```
MINGW64/c/Users/semen/trpp/model_search
LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git add .

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git commit -m "Commit on branch2"
[branch2 6026cf2] Commit on branch2
1 file changed, 1 insertion(+)
create mode 100644 file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git checkout branch1
Switched to branch 'branch1'

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ echo "7525" > file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git add .

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git commit -m "New commit on branch1"
[branch1 ebc6f75] New commit on branch1
1 file changed, 1 insertion(+), 1 deletion(-)

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git checkout branch2
Switched to branch 'branch2'

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$
```

Рисунок 26 – Выполнение задания 4 третьей части

```
MINGW64/c/Users/semen/trpp/model_search
Switched to branch 'branch1'

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ echo "7525" > file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git add .

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git commit -m "New commit on branch1"
[branch1 ebc6f75] New commit on branch1
1 file changed, 1 insertion(+), 1 deletion(-)

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git checkout branch2
Switched to branch 'branch2'

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ echo "16155" > file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git add file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git commit -m "New commit on branch2"
[branch2 9bcc1fc] New commit on branch2
1 file changed, 1 insertion(+), 1 deletion(-)

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$
```

Рисунок 27 – Выполнение задания 4 третьей части

```
MINGW64/c/Users/semen/trpp/model_search

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git add file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git commit -m "New commit on branch2"
[branch2 9bcc1fc] New commit on branch2
1 file changed, 1 insertion(+), 1 deletion(-)

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git checkout branch1
Switched to branch 'branch1'

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ echo "1568" > file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git add file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git commit -m "Final commit on branch1"
[branch1 4213176] Final commit on branch1
1 file changed, 1 insertion(+), 1 deletion(-)

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git checkout branch2
Switched to branch 'branch2'

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$
```

Рисунок 28 – Выполнение задания 4 третьей части

```
MINGW64/c/Users/semn/trpp/model_search
Switched to branch 'branch1'

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ echo "1568" > file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git add file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git commit -m "Final commit on branch1"
[branch1 4213176] Final commit on branch1
1 file changed, 1 insertion(+), 1 deletion(-)

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git checkout branch2
Switched to branch 'branch2'

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ echo "1651" > file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git add file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git commit -m "Final commit on branch2"
[branch2 6dffee1] Final commit on branch2
1 file changed, 1 insertion(+), 1 deletion(-)

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$
```

Рисунок 29 – Выполнение задания 4 третьей части

Слияние двух веток.

Исправление вручную ошибок при слиянии.

```
MINGW64/c/Users/semn/trpp/model_search
$ git commit -m "Final commit on branch2"
[branch2 6dffee1] Final commit on branch2
1 file changed, 1 insertion(+), 1 deletion(-)

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git merge branch1
Auto-merging file.txt
CONFLICT (add/add): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2|MERGING)
$ git nano file.txt
git: 'nano' is not a git command. See 'git --help'.

The most similar commands are
  annotate
  daemon

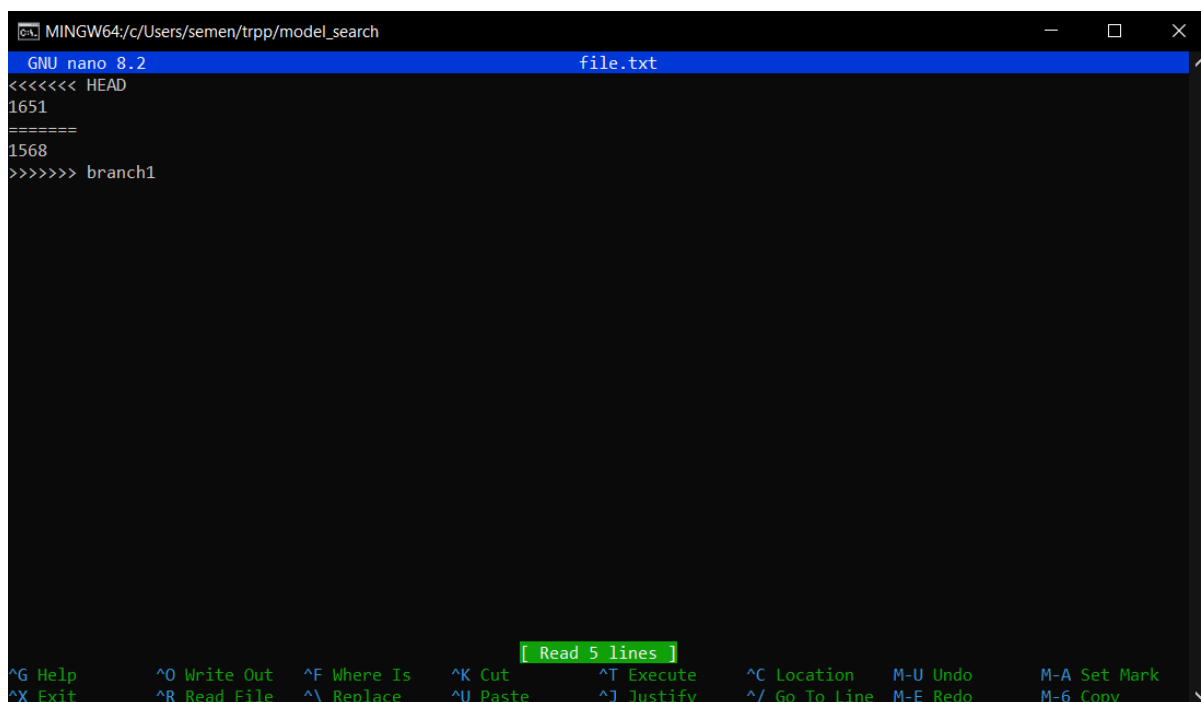
LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2|MERGING)
$ nano file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2|MERGING)
$ cat file.txt
1651

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2|MERGING)
$ git add file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2|MERGING)
$
```

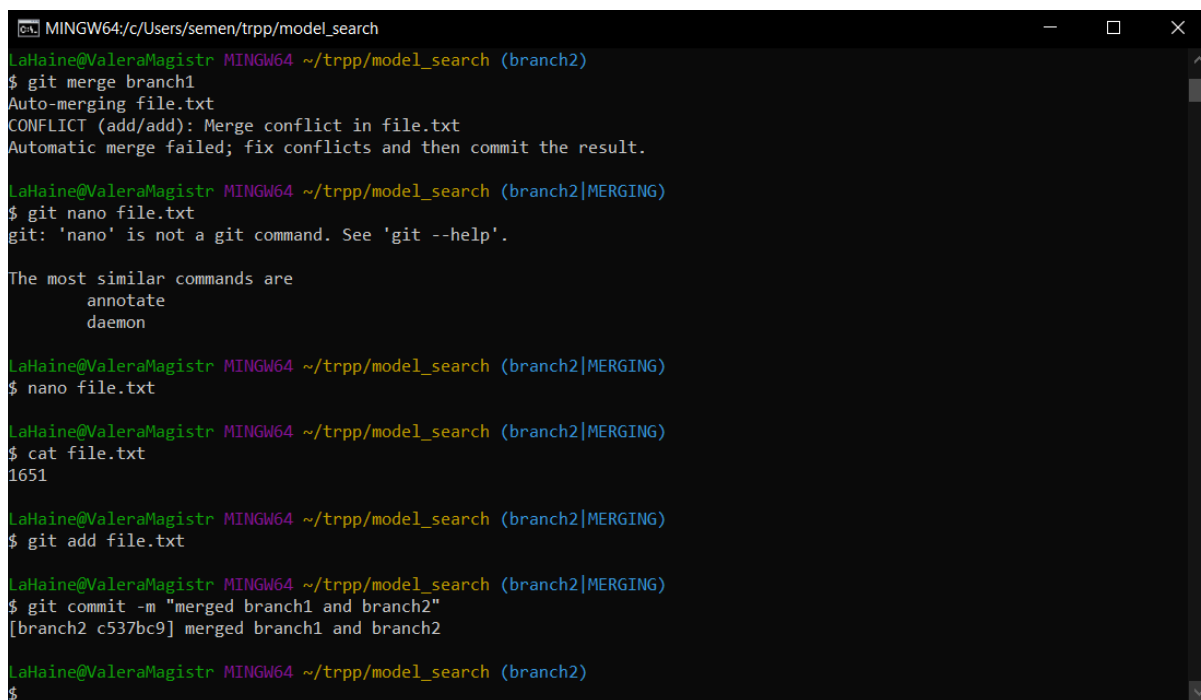
Рисунок 30 – Выполнение задания 5 третьей части



```
MINGW64/c/Users/semn/trpp/model_search
GNU nano 8.2 file.txt
<<<<<<< HEAD
1651
=====
1568
>>>>>>> branch1

^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy
```

Рисунок 31 – Выполнение задания 5 третьей части



```
MINGW64/c/Users/semn/trpp/model_search
LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git merge branch1
Auto-merging file.txt
CONFLICT (add/add): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2|MERGING)
$ git nano file.txt
git: 'nano' is not a git command. See 'git --help'.

The most similar commands are
    annotate
    daemon

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2|MERGING)
$ nano file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2|MERGING)
$ cat file.txt
1651

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2|MERGING)
$ git add file.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2|MERGING)
$ git commit -m "merged branch1 and branch2"
[branch2 c537bc9] merged branch1 and branch2

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$
```

Рисунок 32 – Выполнение задания 5 третьей части

Выгрузка изменений из каждой ветки в удалённый репозиторий.

```
MINGW64/c/Users/semen/trpp/model_search
LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2|MERGING)
$ cat file.txt
1651
LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2|MERGING)
$ git add file.txt
LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2|MERGING)
$ git commit -m "merged branch1 and branch2"
[branch2 c537bc9] merged branch1 and branch2
LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git push origin branch2
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (19/19), 1.60 KiB | 410.00 KiB/s, done.
Total 19 (delta 7), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (7/7), completed with 1 local object.
remote:
remote: Create a pull request for 'branch2' on GitHub by visiting:
remote:   https://github.com/golevss/model_search/pull/new/branch2
remote:
To https://github.com/golevss/model_search.git
 * [new branch]      branch2 -> branch2
LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$
```

Рисунок 33 – Выполнение задания 6 третьей части

```
MINGW64/c/Users/semen/trpp/model_search
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (19/19), 1.60 KiB | 410.00 KiB/s, done.
Total 19 (delta 7), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (7/7), completed with 1 local object.
remote:
remote: Create a pull request for 'branch2' on GitHub by visiting:
remote:   https://github.com/golevss/model_search/pull/new/branch2
remote:
To https://github.com/golevss/model_search.git
 * [new branch]      branch2 -> branch2
LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git checkout branch1
Switched to branch 'branch1'
LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git push origin branch1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'branch1' on GitHub by visiting:
remote:   https://github.com/golevss/model_search/pull/new/branch1
remote:
To https://github.com/golevss/model_search.git
 * [new branch]      branch1 -> branch1
LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$
```

Рисунок 34 – Выполнение задания 6 третьей части

```
MINGW64/c/Users/semn/trpp/model_search
remote:      https://github.com/golevss/model_search/pull/new/branch2
remote:
To https://github.com/golevss/model_search.git
* [new branch]      branch2 -> branch2

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch2)
$ git checkout branch1
Switched to branch 'branch1'

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git push origin branch1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'branch1' on GitHub by visiting:
remote:      https://github.com/golevss/model_search/pull/new/branch1
remote:
To https://github.com/golevss/model_search.git
* [new branch]      branch1 -> branch1

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (master)
$ git push origin master
Everything up-to-date

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (master)
$
```

Рисунок 35 – Выполнение задания 6 третьей части

Повторное клонирование репозитория.

```
MINGW64/c/Users/semn/model_search
To https://github.com/golevss/model_search.git
* [new branch]      branch1 -> branch1

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (branch1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (master)
$ git push origin master
Everything up-to-date

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (master)
$ cd ~

LaHaine@ValeraMagistr MINGW64 ~ (new_branch)
$ git clone https://github.com/golevss/model_search.git
Cloning into 'model_search'...
remote: Enumerating objects: 293, done.
remote: Counting objects: 100% (94/94), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 293 (delta 55), reused 57 (delta 46), pack-reused 199 (from 1)
Receiving objects: 100% (293/293), 293.14 KiB | 615.00 KiB/s, done.
Resolving deltas: 100% (134/134), done.

LaHaine@ValeraMagistr MINGW64 ~ (new_branch)
$ cd model_search/

LaHaine@ValeraMagistr MINGW64 ~/model_search (master)
$
```

Рисунок 36 – Выполнение задания 8 третьей части

Три новых коммита.

```
MINGW64/c/Users/semen/model_search
LaHaine@ValeraMagistr MINGW64 ~/model_search (branch1)
$ touch text1.txt text2.txt text3.txt

LaHaine@ValeraMagistr MINGW64 ~/model_search (branch1)
$ git add text1.txt

LaHaine@ValeraMagistr MINGW64 ~/model_search (branch1)
$ git commit -m "Commit 1"
[branch1 d13ad72] Commit 1
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 text1.txt

LaHaine@ValeraMagistr MINGW64 ~/model_search (branch1)
$ git add text2.txt

LaHaine@ValeraMagistr MINGW64 ~/model_search (branch1)
$ git commit -m "Commit 2"
[branch1 140cb14] Commit 2
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 text2.txt

LaHaine@ValeraMagistr MINGW64 ~/model_search (branch1)
$ git add text3.txt

LaHaine@ValeraMagistr MINGW64 ~/model_search (branch1)
$ git commit -m "Commit 3"
[branch1 d2ffe4e] Commit 3
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 text3.txt
```

Рисунок 37 – Выполнение задания 9 третьей части
Выгрузка всех трёх изменений в новый репозиторий.

```
MINGW64/c/Users/semen/model_search
LaHaine@ValeraMagistr MINGW64 ~/model_search (branch1)
$ git push origin branch1
To https://github.com/golevss/model_search.git
 ! [rejected]        branch1 -> branch1 (non-fast-forward)
error: failed to push some refs to 'https://github.com/golevss/model_search.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. If you want to integrate the remote changes,
hint: use 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

LaHaine@ValeraMagistr MINGW64 ~/model_search (branch1)
$ git pull origin branch1
From https://github.com/golevss/model_search
 * branch            branch1      -> FETCH_HEAD
Merge made by the 'ort' strategy.
 file.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 file.txt

LaHaine@ValeraMagistr MINGW64 ~/model_search (branch1)
$ git push origin branch1
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 837 bytes | 837.00 KiB/s, done.
Total 9 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/golevss/model_search.git
 4213176..3fbb48d branch1 -> branch1
```

Рисунок 38 – Выполнение задания 10 третьей части
Возвращение в прошлый репозиторий, выгрузка изменений.


```
MINGW64/c/Users/semen/trpp/model_search
LaHaine@ValeraMagistr MINGW64 ~/model_search (branch1)
$ cd ~/trpp/model_search/

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (master)
$ touch mew_text.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (master)
$ git add mew_text.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (master)
$ git commit -m "New commit in old repo"
[master 7caf847] New commit in old repo
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mew_text.txt

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (master)
$ git push --force origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 280 bytes | 280.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/golevss/model_search.git
d90bc39..7caf847 master -> master

LaHaine@ValeraMagistr MINGW64 ~/trpp/model_search (master)
$
```

Рисунок 39 – Выполнение задания 11 третьей части

Получение этих изменений в новом репозитории.

```
MINGW64/c/Users/semen/model_search
LaHaine@ValeraMagistr MINGW64 ~/model_search (branch1)
$ git checout master
git: 'checout' is not a git command. See 'git --help'.

The most similar command is
  checkout

LaHaine@ValeraMagistr MINGW64 ~/model_search (branch1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

LaHaine@ValeraMagistr MINGW64 ~/model_search (master)
$ git pull origin master
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 260 bytes | 9.00 KiB/s, done.
From https://github.com/golevss/model_search
* branch      master      -> FETCH_HEAD
d90bc39..7caf847 master    -> origin/master
Updating d90bc39..7caf847
Fast-forward
 mew_text.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mew_text.txt

LaHaine@ValeraMagistr MINGW64 ~/model_search (master)
$
```

Рисунок 40 – Выполнение задания 12 третьей части

КОНТРОЛЬНЫЕ ВОПРОСЫ

4. Что такое коммит?

Коммит — это операция в системе контроля версий (например, Git), которая сохраняет изменения в репозитории, фиксируя текущую версию файлов.

7. Что такое слияние двух веток?

Слияние двух веток (merge) — это процесс, при котором изменения из одной ветки (например, branch1) интегрируются в другую ветку (например, master или branch2). В Git это операция, которая объединяет истории двух веток, создавая новый коммит слияния.

10. Для чего нужен .gitignore?

Для указания git какие файлы не должны индексироваться.

13. Что делает команда git log?

Отображает историю коммитов, включая информацию о этих коммитах. Также имеет множество видов отображения для корректного понимания истории коммитов.

25. Как отменить действие команды "git add" на файл?

Команда *git reset <название файла>* удаляет индексацию указанных файлов (то есть, отменяет добавление файла в коммит), но при этом не затрагивает сами изменения в рабочем каталоге.

19. Чем отличаются команды "git push" и "git pull"?

Команда *git push* отправляет изменения в удаленный репозиторий, а команда *git pull* принимает изменения из удалённого репозитория.

15. Что делает команда git show?

Команда *git show <хэш коммита>* используется для отображения подробной информации о конкретном коммите или объекте в Git.

16. Что делает команда git stash?

Команда `git stash` временно сохраняет все несохранённые изменения в рабочем каталоге, чтобы можно было переключиться на другую ветку или выполнить другую операцию, не теряя текущие наработки.

ВЫВОД

В ходе данной работы была изучена система контроля версий. Изучены основные команды, работа с удалённым репозиторием, проблемы при слиянии веток, работа с несколькими ветками.